REMARKS

The above-identified Office action finally rejected claims 1-12, 28, 38-40, and 42-50 under 35 U.S.C. §103(a) as being allegedly obvious from U.S. Patent No. 5,832,530 to Paknad et al. in view of U.S. Patent No. 5,319,748 to Motoyama. The rejections of record are not in accordance with legal precedent to clearly establish a *prima facie* case of obviousness and, therefore, are respectfully traversed. Accordingly, Applicant respectfully requests that the Application be reconsidered in view of the reasons set forth below.

Rejections Under 35 U.S.C. § 103(a)

Claims 1-12, now pending, stand rejected under 35 U.S.C. § 103(a) as allegedly being obvious over U.S. Patent No. 5,832,530 to Paknad et al. ("Paknad") in view of U.S. Patent No. 5,319,748 to Motoyama ("Motoyama"). For the reasons set forth below, these rejections are respectfully traversed.

To render a claim obvious, the combination of Paknad and Motoyama must: (1) teach or suggest all the claim limitations; (2) provide a reasonable expectation of success; and (3) provide some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the references. MPEP § 2143. The Office action fails to meet these three requirements in alleging that claims 1-12 are obvious from Paknad in view of Motoyama. Specifically, each of the claims in the Application contains one or more limitations that are completely absent from Paknad and Motoyama. To establish *prima facie* obviousness of a claimed invention, all the claim limitations must be taught or suggested by the prior art. *In re Royka*, 490 F.2d 981, 180 USPQ 580 (CCPA 1974). "All words in a claim must be considered in judging the patentability of that claim against the prior art." *In re Wilson*, 424 F.2d 1382, 1385, 165 USPQ 494, 496 (CCPA 1970).

A. **Claim 1**

Independent claim 1 of the present invention is directed to a method of associating a particular path defined in a page description language specification with a plurality of special attributes, comprising the steps of: (a) monitoring a first text string defined by a first page description language text command in the specification for a first special character or a first special string of characters, the first special character or the first special string of characters being indicative of a first special attribute; (b) monitoring a second text string defined by a second page description language text command in the specification for a second special character or a second special string of characters, the second special character or the second special string of characters being indicative of a second special attribute; (c) responsive to a detection of the first special character or the first special string of characters in the first text string, identifying a path defined by a page description language path command and having a predetermined relationship with the first text command in the specification as the particular path associated with the first special attribute; and (d) responsive to a detection of the second special character or the second special string of characters in the second text string, identifying the path defined by the page description language path command and having a predetermined relationship with the second text command in the specification as the particular path associated with the second special attribute.

The Office action contends that Paknad discloses elements (a) and (b) of independent claim 1, namely

> monitoring a first (or second) text string defined by a first (or second) page description language text command in the specification for a first (or second) special character or a first (or second) special string of characters, the first (or second) special character or the first (or second) special string of characters being indicative of a first (or second) special attribute.

In support of this contention, the Office action cites the following excerpt of Paknad:

> A digital computation apparatus stores a page of the document, where each text segment preferably has an

> associated x coordinate and y coordinate which indicate where the text segment is to be displayed on a displayed page. The page includes text segments of one or more characters that have not been identified as words.

(col. 2, ln. 50-55) This explanation from Paknad, however, simply states that each text segment in a portable document has *x* and *y* coordinates that specify where that segment is to be displayed on the page. This feature of documents and graphics is, of course, present in numerous file formats, and it has no relation to the present invention's monitoring of a PDL text command for a special character string whose presence would be indicative of some special attribute, as stated in claim 1.

Paknad does describe interpreting commands of a portable electronic document in order to piece together the text strings in the document. (col. 11, ln. 10-13) It is critically important, however, to realize just what the Paknad method is doing when it interprets the portable electronic document's commands. Each of these commands specifies some portion of the document's content to be displayed (text characters or strings, in the case of the Paknad method's operation) on the page, and the Paknad method reads through these commands to construct the text strings that the commands prescribe. By constructing the text strings prescribed by the document's commands, the Paknad method can compile a list of all the words contained in the document. The critical point is this: As it goes through the document's commands, the Paknad method is not looking for any characters or strings of particular significance (a "special character . . . or string" that is "indicative of a . . . special attribute," in the words of claim 1 of the present invention). The Paknad method is simply assembling strings of characters as instructed by the portable electronic document's commands and determining where the physical spacing between contiguous characters is sufficiently great as to indicate a word break. (col. 2, ln. 63 – col. 3, ln. 7) Thus, the Paknad method is constantly determining the spacing between characters, but none of the characters it is reading carry any special significance or "attributes."

By contrast, the method of claim 1 of present invention reads text commands from a PDL file *specifically looking for a "special character or . . . string of characters" that is "indicative of a . . . special attribute."* For example, this special attribute can be an association with particular merge file containing data to be inserted into the document. As seen in the exemplary embodiment, when the method detects the "special character or string" that has a predefined association with a merge file, this instructs the method to operate on the contents of that merge file. (Application, p.19, ln.28 – p.20, ln.23) The method of claim 1 is monitoring command strings, *looking for special characters or strings that have special attributes*. This feature is completely absent from Paknad, which does not look for specific characters or strings at all. Accordingly, the first and second elements of claim 1 are not disclosed by Paknad.

The Office action further contends that elements (c) and (d) of claim 1 would have been obvious from Paknad in view of Motoyama. The third and fourth elements of claim 1 are:

> Responsive to a detection of the first (second) special character or the first (second) special string of characters in the first (second) text string, identifying a path defined by a page description language path command and having a predetermined relationship with the first (second) text command in the specification as the particular path associated with the first (second) special attribute.

The Office action contends that these steps would have been obvious in view of Motoyama's teaching of a hierarchy of pagesets defined by pointers to locations in a stack memory.

As an initial matter, even if this teaching of Motoyama were relevant to the present invention, the method of claim 1 would not have been obvious because, as discussed above, the first two elements (elements (a) and (b)) of claim 1 are absent from Paknad. Moreover, elements (c) and (d) are not obvious in view of Motoyama's reference to pageset pointers, which simply provide a means for demarcating and locating the boundaries between pages, pictures, or other elements in a document. Motoyama does not teach anything about "*identifying*" a path (or other attribute) "having a *predetermined relationship*" with a "*text command*", as required by elements (c) and (d). This step would not have been obvious from Paknad or Motoyama because neither of those references is concerned with monitoring commands for special characters having predetermined relationship to a special command or data attribute, as the present invention is.

For the foregoing reasons, Applicant respectfully submits that claim 1 is in condition for allowance and that the rejection of record by withdrawn.

B. <u>Claims 2-4</u>

Claims 2-4 depend from claim 1 and are therefore allowable for at least the same reasons as provided above for claim 1. Of particular relevance to claims 2 and 3 is the fact, discussed above, that Paknad does not address any "predetermined relationship" of commands to one another, as required by claims 2 and 3. Instead, the Paknad method simply reads commands in a portable electronic document and constructs the character strings prescribed by those commands in order to form words.

Regarding claim 4, as discussed above, neither Paknad nor Motoyama addresses any "special attribute . . . associated with a first (or second) merge file," as required by claim 4. As an initial matter, neither reference discusses "merge files," which are defined in the present application to be a file containing data (such as text) that is to be inserted (or "merged") into a template bitmap for a page from a PDL file. (Application, p.6, ln. 24-26) As explained above, Paknad is concerned with identifying words in a portable electronic document, and Motoyama is concerned with a means for demarcating and locating the boundaries between pages, pictures, or other elements in a document such that specific pages, pictures, or elements can be located quickly and accessed directly. Neither reference contains any data structure that is congruent to, or performs a similar function to, a merge file of the present invention. Furthermore, the excerpt from

Motoyama cited in the Office action, does not support claim 4. This excerpt from Motoyama states:

> The tokensequence which contain [sic] specific tokens or commands for defining specific images along with necessary operators is called Content while other elements are called Structure. The Structure sets up the environment for Content to generate the appropriate output images.

(col. 3, ln. 40-45) This passage simply states, as the Office action correctly points out, that a document can contain commands providing structure to the content of the document. It does not, however, address associating any "special attribute" indicated by a special string of characters with a merge file or any other item. Accordingly, the method of claim 4 would not have been obvious in view of Motoyama.

C. Claim 5

Independent claim 5 of the present invention is directed to a method for wrapping data to an arbitrary path defined by a page description language, comprising the steps of: (a) designating a path defined in a page description language specification as a wrapping path, the wrapping path having a wrapping-path boundary; (b) processing the specification to produce a template bitmap, the template bitmap being a bitmap or raster-data representation of a template image defined by the specification; (c) associating a block of text with the wrapping path; (d) associating an external bitmap with the wrapping path; (e) merging the external bitmap into the template bitmap, the external bitmap having an external-bitmap boundary; (f) adding the external-bitmap boundary to the wrapping-path boundary, forming a composite boundary; and (g) merging bitmap representations of the text from the block of text, according to the composite boundary and according to predefined flow rule, into the template bitmap to create a merged bitmap.

The Office action contends that each element of claim 5 is disclosed in Paknad or would have been obvious from Paknad in view of Motoyama. Element (a), "designating a path defined in a page description language specification as a wrapping path, the wrapping path having a wrapping-path boundary," is not suggested or rendered obvious in view of Motoyama. As discussed above, Motoyama is concerned with a means for demarcating and locating the boundaries between pages, pictures, or other elements in a document such that specific pages, pictures, or elements can be located quickly and accessed directly. It does not discuss or suggest designating an area on a page template as a "path" into which text or graphics will be inserted, and it does not discuss designating any such "path" area as a "wrapping path" into which text can be flowed. These text-handling features of the present invention and claim 5 are nowhere suggested by Motoyama.

Element (b) of claim 5, "processing the specification to produce a template bitmap, the template bitmap being a bitmap or raster-data representation of a template image defined by the specification," is not taught by Paknad because Paknad does not produce a "template" of the static items on a page that is rasterized separately from any variable data on the page. Because this second element of claim 5 produces a "template bitmap" of the "template image" and not simply a bitmap of "character codes" as seen in Paknad, it is not taught by Paknad.

Element (c) of claim 5, "associating a block of text with the wrapping path," is not taught by Paknad, which makes absolutely no mention or suggestion of *paths* or *wrapping* paths. These concepts are nowhere to be found in Paknad, which is concerned with reading commands in a portable electronic document and constructing the character strings prescribed by those commands in order to form words.

Element (d) of claim 5, "associating an external bitmap with the wrapping path," is not taught by Paknad. The Office action quotes the following portion of Paknad in support:

> In addition, at least portions of adjacent rotated text objects are added as a word to the word list by the word identifying mechanism when bounding boxes of the text objects intersect or are separated by a threshold gap distance.

(col. 3, ln. 7-11) While this word recognition procedure does "teach[] a boundary" applied to text objects on a page as noted by the Office action, it does not pertain to external bitmaps or a wrapping path, as required by this element of claim 5. Indeed, as noted above, the concept of a wrapping path is completely absent from Paknad.

Element (e) of claim 5, "merging the external bitmap into the template bitmap, the external bitmap having an external-bitmap boundary," is omitted from the Office action's discussion. For the same reasons as stated above with respect to the foregoing elements, Paknad does not teach this element. It must be emphasized that Paknad does not perform any merging or inserting of bitmaps or data objects; Paknad is concerned solely with *reading commands from* a portable electronic document and constructing the character strings prescribed by those commands in order to form words. Paknad does not *create* any pages, bitmaps, templates, or other media.

Elements (f) and (g) of claim 5, "adding the external-bitmap boundary to the wrapping-path boundary, forming a composite boundary," and "merging bitmap representations of the text from the block of text, according to the composite boundary and according to a predefined flow rule, into the template bitmap to create a merged bitmap," are not taught or suggested by Paknad. The excerpt of Paknad cited by the Office action (col. 11, ln. 20-27), like the excerpt from Paknad's summary quoted above, describes a means for determining whether two text segments are in sufficiently close proximity on the page to be part of the same word. Paknad does not *manipulate* or *merge* text and bitmaps; it simply *reads* commands to construct the character strings prescribed by those commands in order to form words. As with claim 4, the cited excerpt from Motoyama (col. 3, ln. 40-45) does not suggest the method of claim 5 because it simply states that a document can contain commands providing structure to the content of the document. Neither Motoyama nor Paknad suggests merging bitmap representations of a block of text into a template bitmap for a page, as claim 5 requires. Indeed, neither Motoyama nor Paknad addresses the *creation* of any pages, bitmaps, templates, or other media at all; these references are solely concerned with *reading* and *interpretation* of data in a previously created file.

Finally, it must be emphasized that obviousness of an invention is not to be ascertained on an element-by-element basis. A claim is rendered obvious by the prior art only if the claimed structure or method, *viewed as a whole*, would have been obvious to one of ordinary skill in the art from the prior art. *See* MPEP § 2141.02 ("In determining the differences between the prior art and the claims, the question under 35 U.S.C. 103 is not whether the differences themselves would have been obvious, but whether the claim invention as a whole would have been obvious.") (emphasis in original) (*citing Stratoflex, Inc. v. Aeroquip Corp.*, 713 F.2d 1530 (Fed. Cir. 1983); *Schenck v. Nortron Corp.*, 713 F.2d 782 (Fed. Cir. 1983)). Applying this standard, there can be no doubt that the method of claim 5 would NOT have been obvious from the cited prior art, which is completely devoid of many of the required features and concepts essential to the claim 5 method, as pointed out above.

D. Claim 6

Claim 6 depends from claim 5 and is therefore allowable for at least the same reasons as provided above for claim 5.

E. Claim 7

Independent claim 7 of the present invention is directed to a method for wrapping data to an arbitrary path defined by a page description language, comprising the steps of: (a) designating a path defined in a page description language specification as a wrapping path, the wrapping path having a boundary; (b) defining a first graphics state for the path; (c) defining a second graphics state for the path; (d) processing the specification to produce a template bitmap, the template bitmap being a bitmap or raster-data representation of a template image defined by the specification; (e) associating a text file with the wrapping path, the text file including a first block of text separated from a second block of text by a field delimiter; (f) creating first bitmap representations of the first block of text by applying the first graphics state to the first block of text; (g) merging the first bitmap representations of the text, according to the boundary and according to a predefined flow rule, into the template; (h) creating second bitmap representation of the second block of text by applying the second graphics state to the second block of text; and (i) merging the second bitmap representation of the text, according to the boundary and according to the predefined flow rule, into the template bitmap.

The Office action contends that each element of claim 7 is disclosed in Paknad or would have been obvious from Paknad in view of Motoyama. Claim 7 contains several steps that are similar to steps in claim 5, and these steps are not taught or suggested by Paknad or Motoyama for the same reasons stated above. Particularly, the concepts of a "path" area and a "wrapping path" into which text can be flowed are completely absent from Paknad and Motoyama.

Element (a) of claim 7, "designating a path defined in a page description language specification as a wrapping path, the wrapping path having a boundary," is not suggested or rendered obvious in view of Motoyama for the same reasons as stated above regarding the (very similar) first element of claim 5.

Element (d) of claim 7, "processing the specification to produce a template bitmap, the template bitmap being a bitmap or raster-data representation of a template image defined by the specification," is identical to the second element of claim 5 and is not taught by Paknad for the same reasons stated above.

Element (e) of claim 7, "associating a text file with the wrapping path, the text file including a first block of text separated from a second block of text by a field delimiter," is not taught by Paknad because Paknad does not use text files; the Paknad specification expressly contrasts ASCII text files with the portable electronic document files (such as Adobe PDF format) that it uses (col. 8, ln. 2-6), which contain commands for displaying text and other objects on a page. Additionally, as discussed above, the concept of a "wrapping path," which is an essential part of this element, is completely absent from Paknad.

Elements (g) and (i) of claim 7, "merging the first (second) bitmap representations of the text, according to the boundary and according to the predefined flow rule, into the template bitmap," are not taught by Paknad because Paknad does not address the task of merging or the concept of a "flow rule" for inserting text, both of which are essential to this element. The Office action contends that the task of merging one bitmap into another bitmap would have been obvious in view of Motoyama. However, this argument is incorrect because the cited passage from Motoyama (col. 3, ln. 40-45) simply states that a document can contain commands providing structure to the content of the document. It says absolutely nothing to suggest the task of merging one bitmap into another "template" bitmap, which is essential to these element of claim 7.

F. Claim 8

Claim 8 contains many of the same or very similar elements as claim 7 and is therefore allowable for at least the same reasons as provided above for claim 7. Additionally, one unique element of claim 8, "replacing all occurrences of a predetermined word in the text block with a substitute word," is completely absent from Paknad and Motoyama.

G. Claim 9

Independent claim 9 of the present invention is directed to a method for wrapping data to an arbitrary path defined by a page description language, comprising the steps of: (a) designating a path defined in a page description language specification as a wrapping path; the wrapping path having a boundary; (b) defining a graphics state for the path; (c) processing the specification to produce a template bitmap, the template bitmap being a bitmap or raster-data representation of a template image defined by the specification; (d) associating a text block with the wrapping path, the text block including a plurality of words and a delimiter; (e) creating bitmap representations of the text block by applying the graphics state to the text block; and (f) merging the bitmap representations of the text block, according to the boundary, according to predefined flow rule and according to the delimiter, into the template.

The Office action contends that each element of claim 9 is disclosed in, or would have been rendered obvious by, Paknad or Motoyama. Claim 9 contains several steps that are similar to steps in claim 7, and these steps are not taught or suggested by Paknad or Motoyama for the same reasons stated above. Particularly, the concepts of a "path" area and a "wrapping path" into which text can be flowed are completely absent from Paknad and Motoyama.

Element (a) of claim 9, "designating a path defined in a page description language specification as a wrapping path, the wrapping path having a boundary," is not suggested or rendered obvious in view of Motoyama for the same reasons as stated above regarding the (very similar) first element of claims 5 and 7.

Element (c) of claim 9, "processing the specification to produce a template bitmap, the template bitmap being a bitmap or raster-data representation of a template image defined by the specification," is identical to the second element of claim 5 and the fourth step of claim 7 and is not taught by Paknad for the same reasons stated above.

Element (d) of claim 9, "associating a text block with the wrapping path, the text block including a plurality of words and a delimiter," is not taught by Paknad because, as discussed above, the concept of a "wrapping path," which is an essential part of this element, is completely absent from Paknad.

Element (f) of claim 9, "merging the bitmap representations of the text block, according to the boundary, according to the predefined flow rule and according to the delimiter, into the template," is similar to the seventh and ninth elements of claim 7. For the same reasons as stated above in the discussion of claim 7, this element is not anticipated or rendered obvious by Paknad and Motoyama. These references do not address the task of merging or the concept of a "flow rule" for inserting text, both of which are essential to this element. The task of "merging" is not rendered obvious by Motoyama, which says absolutely nothing to suggest the task of merging one bitmap into another "template" bitmap, which is essential to this element.

H. Claims 10 and 11

Claims 10 and 11 depend from claim 9 and are therefore allowable for at least the same reasons as provided above for claim 9. The added limitation in each of claims 10 and 11 both require the task of "merging" one bitmap into another "template" bitmap which, as explained above, is not taught by either cited reference and is not rendered obvious in view of Motoyama, which simply states that a document can contain commands providing structure to the content of the document.

I. Claim 12

Independent claim 12 of the present invention is directed to a method for wrapping data to an arbitrary path defined by a page description language, comprising the steps of: (a) designating a path defined in a page description language specification as a wrapping path, the wrapping path having a wrapping-path boundary; (b) defining a graphics state for the path; (c) processing the specification to produce a template bitmap, the template bitmap being a bitmap or raster-data representation of a template image defined by the specification; (d) saving the template bitmap in memory; (e) associating a block of text with the wrapping path; (f) creating bitmap representations of the block of text by applying the graphics state to the block of text; (g) retrieving a first copy of the template bitmap from memory; (h) merging the bitmap representations of the block of text, according to the boundary and according to the predefined flow rule, into the first copy of the template until an end of the boundary is reached; (i) upon reaching the end of the boundary, retrieving a next copy of the template bitmap from memory; and (j) merging a remainder of the bitmap representations of the block of text, according to the boundary and according to the predefined flow rule, into the next copy of the template.

The Office action contends that each element of claim 12 is disclosed in, or would have been rendered obvious by, Paknad or Motoyama. Claim 12 contains several steps that are similar to steps in claims 5, 7, and 9, and these steps are not taught or suggested by Paknad or Motoyama for the same reasons stated above. Particularly, the concepts of a "path" area and a "wrapping path" into which text can be flowed are completely absent from Paknad and Motoyama.

Element (a) of claim 12, "designating a path defined in a page description language specification as a wrapping path, the wrapping path having a wrapping-path boundary," is not suggested or rendered obvious in view of Motoyama for the same reasons as stated above regarding the identical first element of claim 5.

Element (c) of claim 12, "processing the specification to produce a template bitmap, the template bitmap being a bitmap or raster-data representation of a template image defined by the specification," appears in identical form in claims 5, 7, and 9 and is not taught by Paknad for the same reasons stated above.

Element (e) of claim 12, "associating a block of text with the wrapping path," is identical to the third step of claim 5 and is not taught by Paknad for the same reasons stated above.

Element (h) of claim 12, "merging the bitmap representations of the block of text, according to the boundary and according to the predefined flow rule, into the first copy of template until an end of the boundary is reached" is similar to elements appearing in claims 7 and 9. For the same reasons as stated above, this element is not anticipated or rendered obvious by Paknad and Motoyama. These references do not address the task of merging or the concept of a "flow rule" for inserting text, both of which are essential to this element. The task of "merging" is not rendered obvious by Motoyama, which says absolutely nothing to suggest the task of merging one bitmap into another "template" bitmap, which is essential to this element.

Conclusion

As a final remark, it is worth emphasizing yet again that neither Paknad nor Motoyama contains any suggestion of, or motivation for, several essential features of the present invention, most notably *paths*, *wrapping* paths, *templates*, and *merging* one bitmap into another. Paknad reads commands in a portable electronic document and constructs the character strings prescribed by those commands in order to form words, and Motoyama provides a means for demarcating and locating the boundaries between pages, pictures, or other elements in a document such that specific pages, pictures, or elements can be located quickly and accessed directly. Because several concepts essential to the claims of the present invention are *completely absent* (by reference or inference) from Paknad and Motoyama, there is no question that the claims now pending are in condition for allowance. "To establish *prima facie* obviousness of a claimed invention, all the claim limitations must be taught or suggested by the prior art." MPEP § 2143.03 (*citing In re Royka*, 490 F.2d 981, 180 USPQ 580 (CCPA 1974)).

In light of the foregoing, it is respectfully submitted that claims 1-12, now pending, are distinguishable from the references cited, and in condition for allowance. Reconsideration and withdrawal of the objections and rejections of record is respectfully requested.

If the Examiner wishes to discuss any aspect of this response, please contact the undersigned at the telephone number provided below.

Respectfully submitted,

David A. Mancino
Reg. No. 39,289

30074
Taft, Stettinius & Hollister LLP
425 Walnut Street; Suite 1800
Cincinnati, Ohio 45202-3957
mancino@taftlaw.com
(513) 357-9331